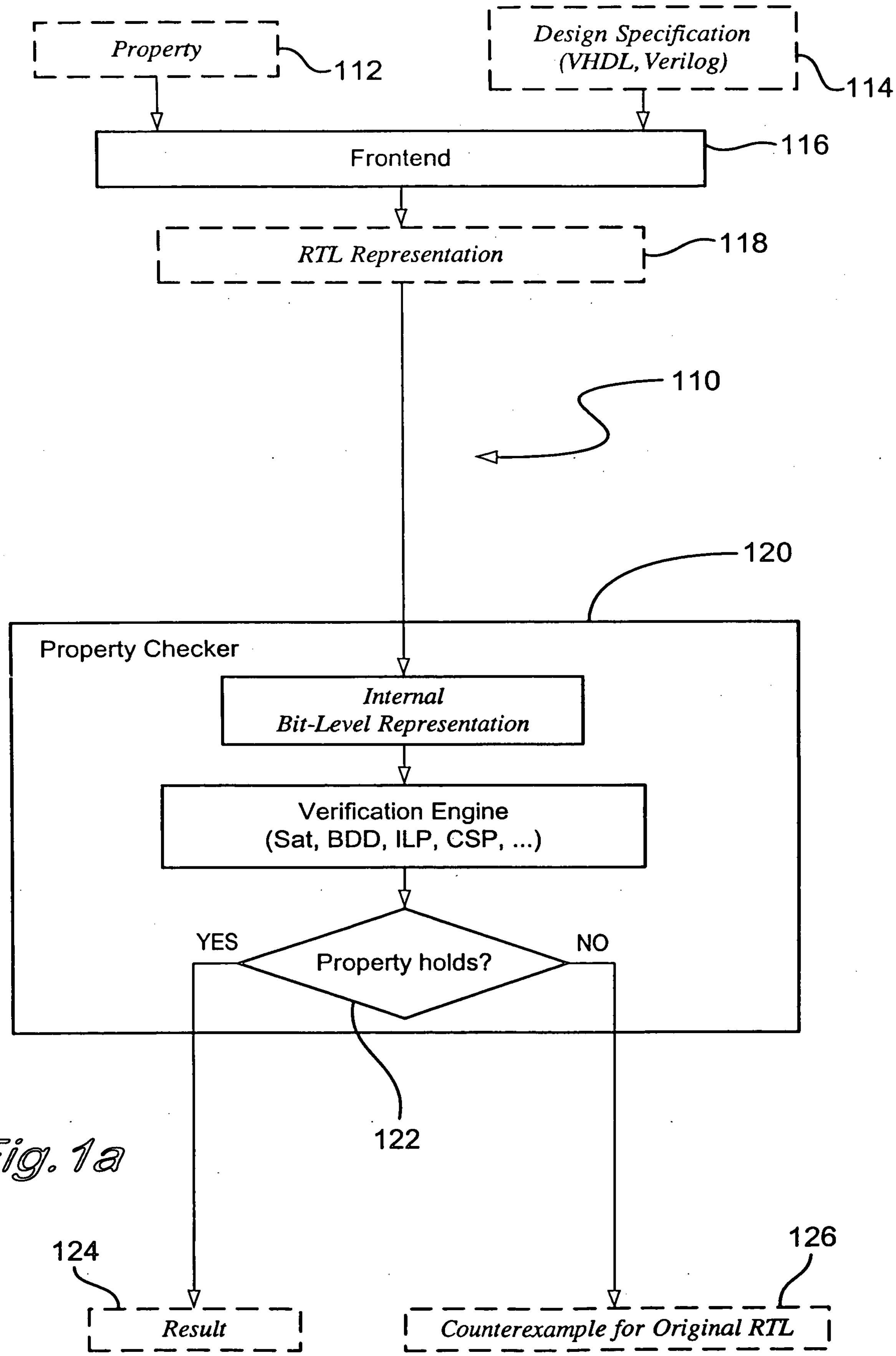

Property
112
Design Specification
(VHDL, Verilog)
114
Frontend
116
RTL Representation
118
110
120
Property Checker
Internal
Bit-Level Representation
Verification Engine
(Sat, BDD, ILP, CSP, ...)
YES
Property holds?
NO
122
124
Result
126
Counterexample for Original RTL

Fig. 1a

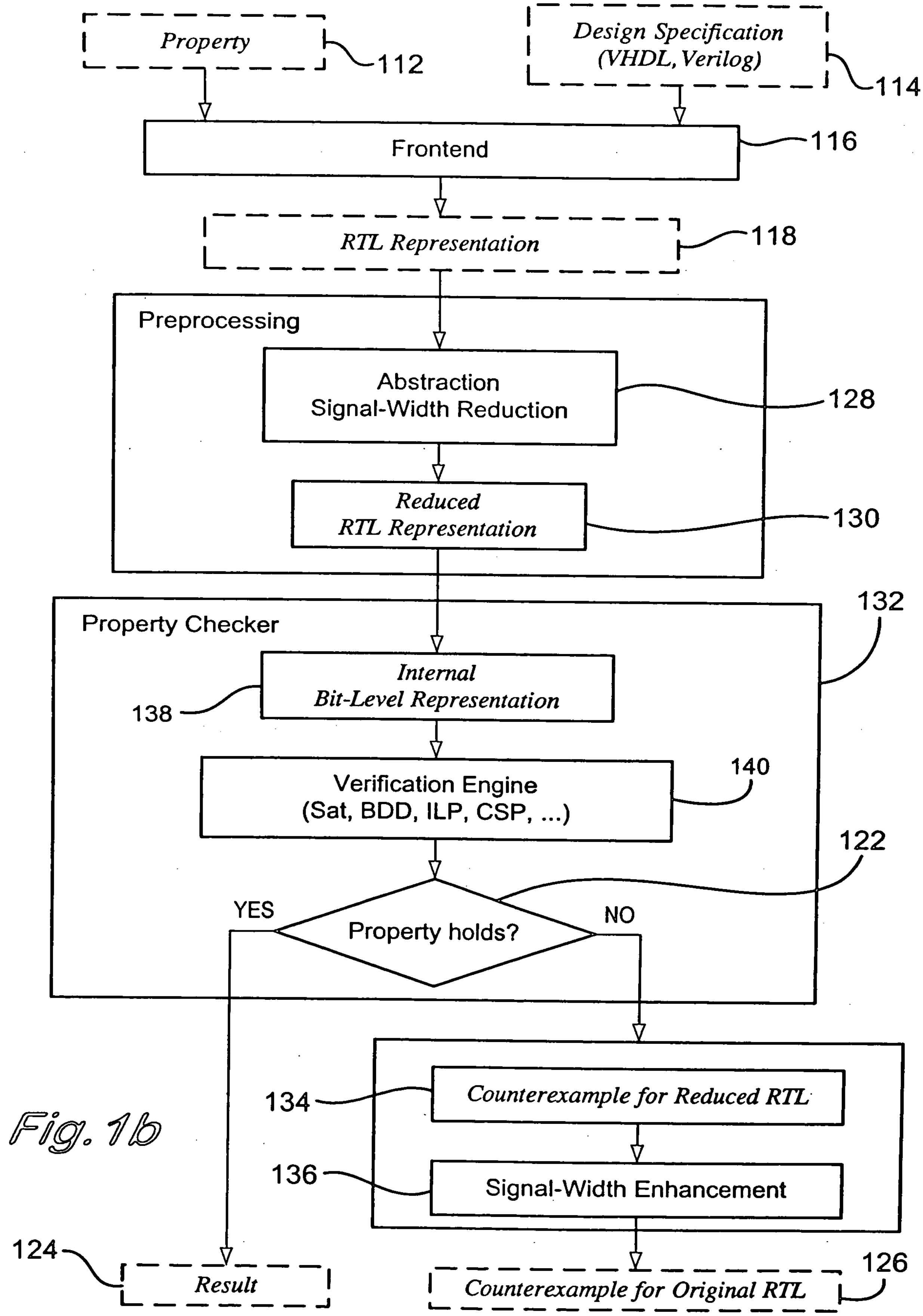

Property
112
Design Specification
(VHDL, Verilog)
114
Frontend
116
RTL Representation
118
Preprocessing
Abstraction
Signal-Width Reduction
128
Reduced
RTL Representation
130
Property Checker
132
Internal
Bit-Level Representation
138
Verification Engine
(Sat, BDD, ILP, CSP, ...)
140
122
YES
Property holds?
NO
Counterexample for Reduced RTL
134
Signal-Width Enhancement
136
124
Result
Counterexample for Original RTL
126

Fig. 1b

2001 P 06030 EP
01 108 653.5

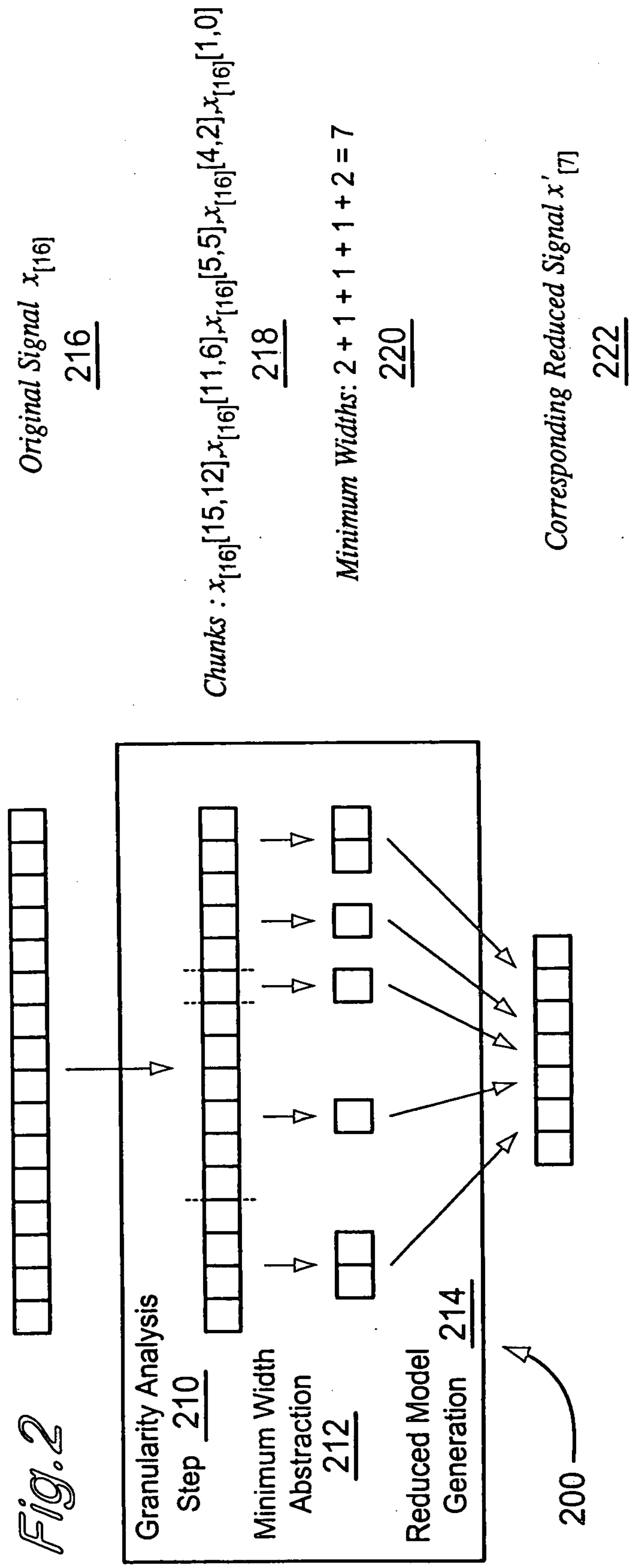


Fig. 2

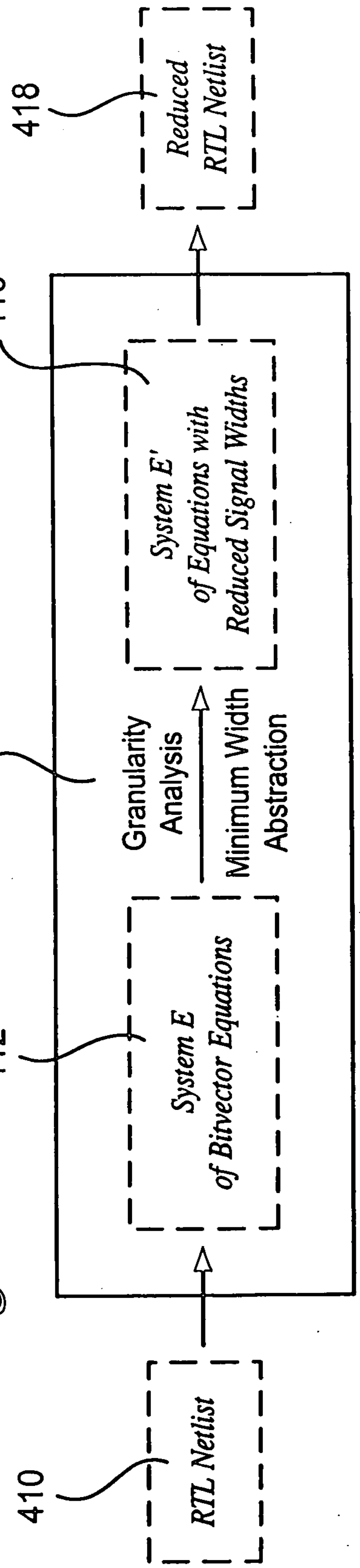


Fig. 4

Fig.3

Bitvector Operator	Syntax	Example
bitvector variables bitvector constants	$x_{[n]}$ $c_{[m]}$	$x_{[8]}, y_{[1]}, z_{[4]}, \ldots$ $10011_{[5]}, 00111111_{[8]}, 0_{[1]}, 1_{[1]}, \ldots$
concatenation	$\otimes$	$x_{[16]} \otimes y_{[4]}$
extraction	$[j, i]$	$x_{[8]}[5, 2]$
bitwise negation (inversion)		neg $(x_{[8]})$
bitwise Boolean operations	and, or, xor nand, nor, xnor	$x_{[12]}$ and $y_{[12]}$, $x_{[12]}$ or $y_{[12]}$, $x_{[12]}$ xor $y_{[12]}$ $x_{[12]}$ nand $y_{[12]}$, $x_{[12]}$ nor $y_{[12]}$, $x_{[12]}$ xnor $y_{[12]}$
if-then-else	ite	ite$(a_{[4]} = b_{[4]}, x_{[8]}, y_{[8]})$ ite$(a_{[4]} < b_{[4]}, x_{[8]}, y_{[8]})$
arithmetic	$+, -$ $*$	$x_{[32]} + y_{[32]}, x_{[32]} - y_{[32]}$ $x_{[16]} * y_{[16]}$
memory read memory write	$mem_{[m \cdot n]}[i_{[l]}]$	$x_{[10]} := mem_{[128 \cdot 10]}[i_{[7]}]$ $mem_{[32 \cdot 8]}[i_{[5]}] := x_{[8]}$

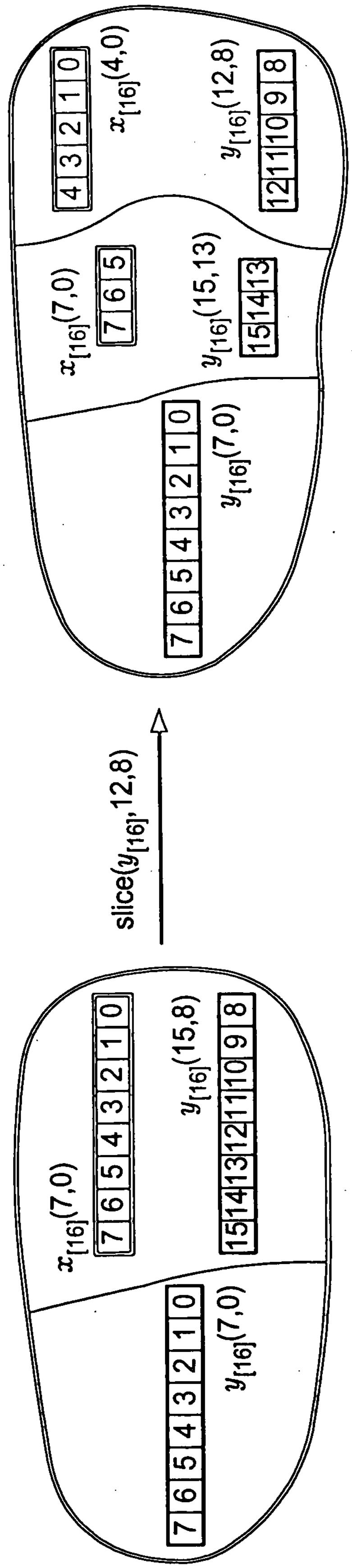

$x_{[16]}(7,0)$
$y_{[16]}(7,0)$
$y_{[16]}(15,8)$
slice($y_{[16]}$,12,8)
$x_{[16]}(7,0)$
$y_{[16]}(15,13)$
$y_{[16]}(7,0)$
$x_{[16]}(4,0)$
$y_{[16]}(12,8)$

Fig.5

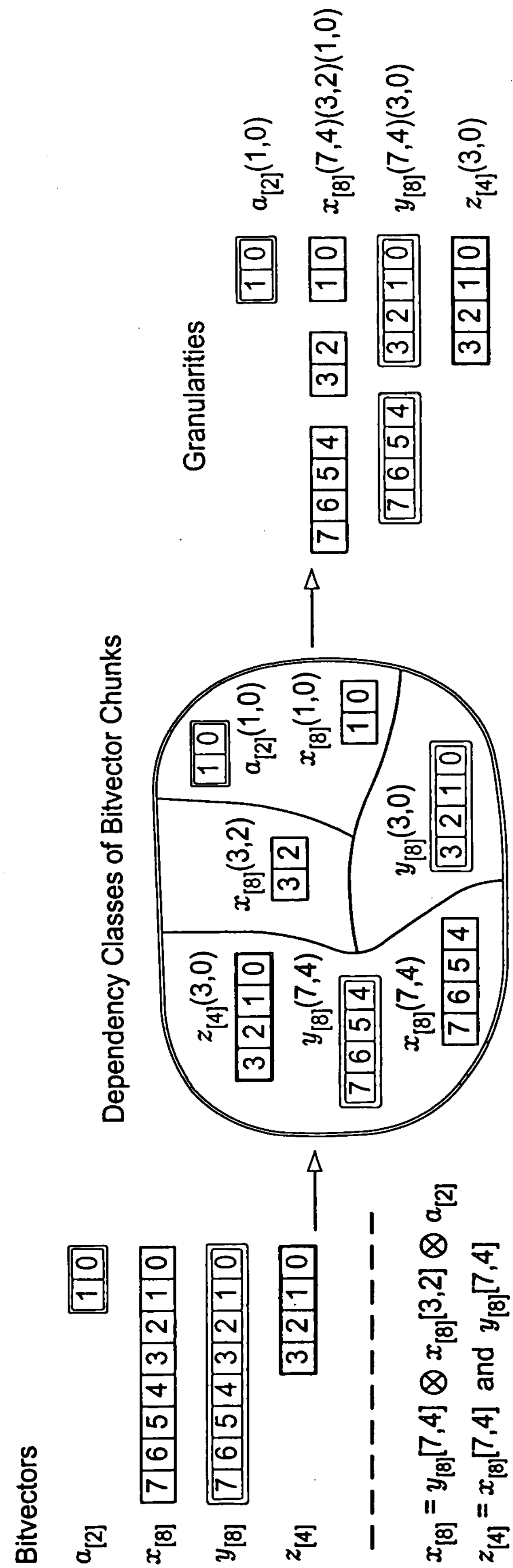

Bitvectors
$a_{[2]}$
$x_{[8]}$
$y_{[8]}$
$z_{[4]}$
Dependency Classes of Bitvector Chunks
$z_{[4]}(3,0)$
$y_{[8]}(7,4)$
$x_{[8]}(7,4)$
$x_{[8]}(3,2)$
$y_{[8]}(3,0)$
$a_{[2]}(1,0)$
$x_{[8]}(1,0)$
Granularities
$a_{[2]}(1,0)$
$x_{[8]}(7,4)(3,2)(1,0)$
$y_{[8]}(7,4)(3,0)$
$z_{[4]}(3,0)$
$x_{[8]} = y_{[8]}[7,4] \otimes x_{[8]}[3,2] \otimes a_{[2]}$
$z_{[4]} = x_{[8]}[7,4]$ and $y_{[8]}[7,4]$

Fig.6

Fig. 6a

Process 1 Granularity Analysis of Bitvector Equations

$gran(e)$ {	
switch (e);	
case $e \equiv$ '$s_{[n]} = t_{1[m_1]} \otimes t_{2[m_2]}$' :	gran('$s_{[n]}[n-1, m_2] = t_{1[m_1]}$'); gran('$s_{[n]}[m_2-1, 0] = t_{2[m_2]}$');
case $e \equiv$ '$s_{[n]} = (t_{1[m_1]} \otimes t_{2[m_2]})[j, i]$' :	**if** $(j < m_2)$ {
	gran('$s_{[n]} = t_{2[m_2]}[j, i]$');
	} **else if** $(i \geq m_2)$ {
	gran('$s_{[n]} = t_{1[m_1]}[j-m_2, i-m_2]$');
	} **else** {
	gran('$s_{[n]} = t_{1[m_1]}[j-m_2, 0] \otimes t_{2[m_2]}[m_2-1, i]$');
	}
case $e \equiv$ '$s_{[n]} = (t_{[m]}[l,k])[j,i]$' :	gran('$s_{[n]} = t_{[m]}[k+j, k+i]$');
case $e \equiv$ '$s_{[n]} = \mathrm{ite}(a_{[m]} = b_{[m]}, t_{t[n]}, te_{[n]})[j,i]$' :	gran('$s_{[n]} = \mathrm{ite}(a_{[m]} = b_{[m]}, t_{t[n]}[j,i], t_{e[n]}[j,i])$');
case $e \equiv$ '$s_{[n]} = (t_{1[m]}$ and $t_{2[m]})[j, i]$' :	gran('$s_{[n]} = t_{1[m]}[j,i]$ and $t_{2[m]}[j, i]$');
...	
cas $e \equiv$ '$s_{[n]} = \mathrm{neg}(t_{[n]})$' :	gran('$s_{[n]} = t_{[n]}$');
case $e \equiv$ '$s_{[n]} = \mathrm{ite}(a_{[m]} = b_{[m]}, t_{t[n]}, t_{e[n]})$' :	gran('$a_{[m]} = b_{[m]}$'); gran('$s_{[n]} = t_{t[n]}$'); gran('$s_{[n]} = t_{e[n]}$');
case $e \equiv$ '$s_{[n]} = t_{1[n]}$ and $t_{2[n]}$' :	gran('$s_{[n]} = t_{1[n]}$'); gran('$s_{[n]} = t_{2[n]}$');
...	
case $e \equiv$ '$x_{[n]}[j,i] = y_{[m]}[l,k]$' :	$slice(x_{[n]}, j,i)$; $slice(y_{[m]}, l,k)$; $join(x_{[n]}\langle j,i\rangle, y_{[m]}\langle l,k\rangle)$;
}	

Fig. 7

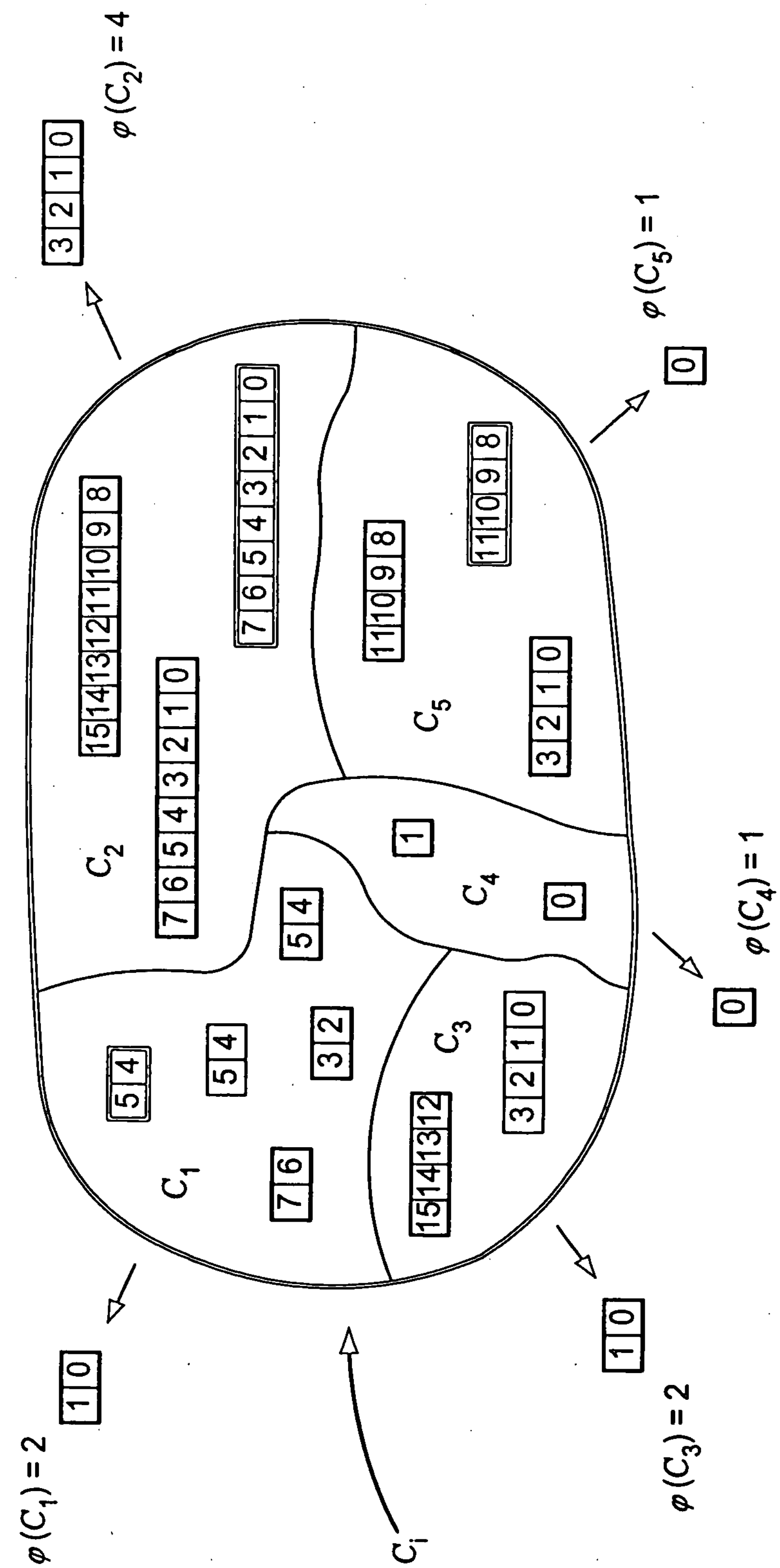

$\varphi(C_2) = 4$
$\varphi(C_5) = 1$
$\varphi(C_4) = 1$
$\varphi(C_3) = 2$
$\varphi(C_1) = 2$
C_1
C_2
C_3
C_4
C_5
C_i

Fig. 7a

Process 2 Reduced Model Generation

for each bitvector variable $x_{[n]}$ {

  $m := 0$;

  for each chunk $x_{[n]}\langle j, i\rangle$ of the computed granularity of $x_{[n]}$ {

    $C := \mathit{find}\,(\, x_{[n]}\langle j, i\rangle\,)$;   // equivalence class containing $x_{[n]}\langle j,i\rangle$

    $m := m + \varphi(C)$;

  }

  if $(m \geq n)$ **then** $m := n$;

  replace all occurrences of $x_{[n]}$ in the bitvector equations by $x'_{[m]}$;

  and adjust all extraction expressions affected by $x_{[n]}$;

}

Fig.8

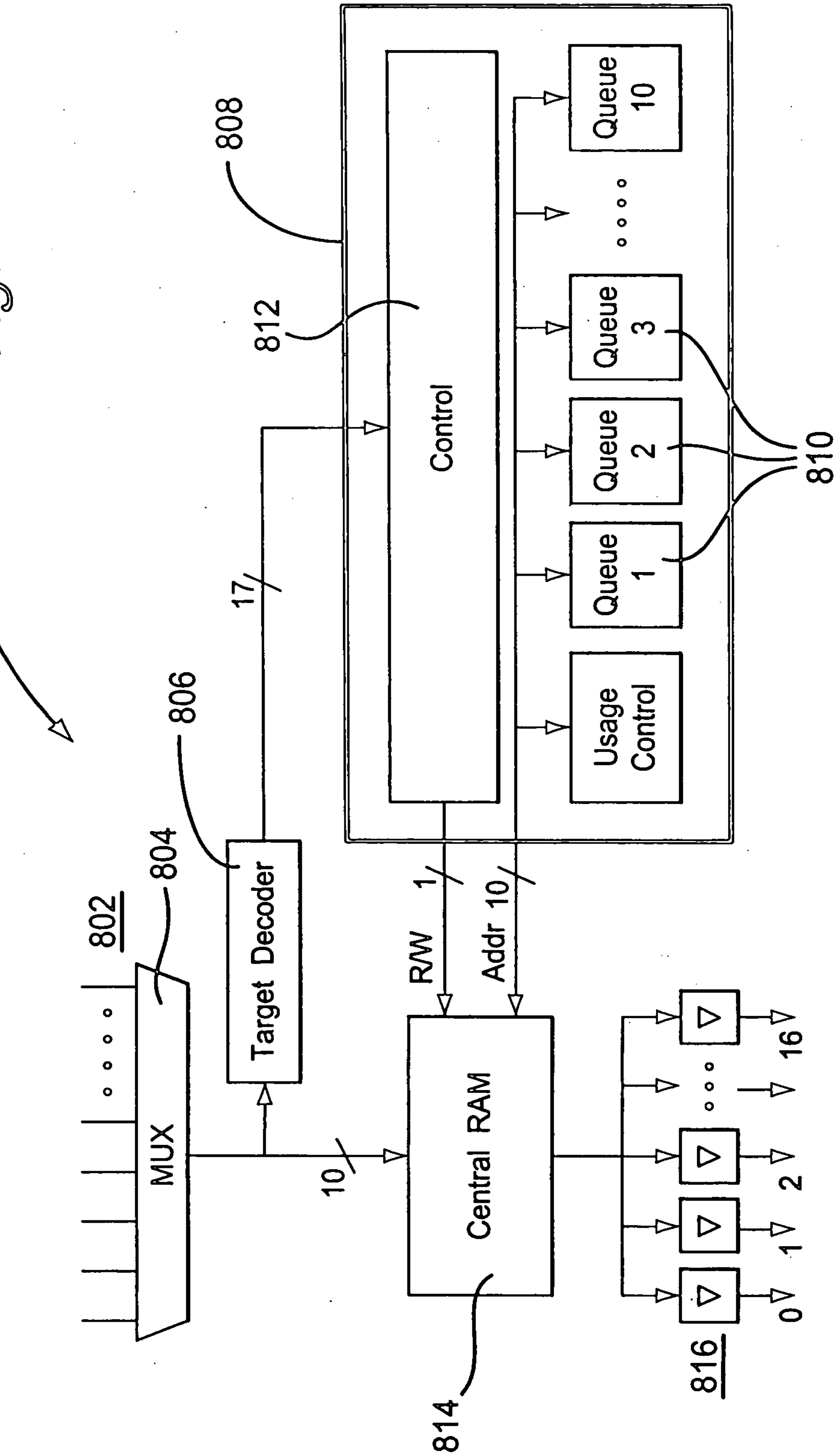

800
802
804
MUX
806
Target Decoder
17
808
812
Control
Queue 10
Queue 3
Queue 2
Queue 1
Usage Control
810
R/W
1
Addr 10
10
Central RAM
814
816
0
1
2
16

Fig.9

	Property	Original design	Reduced model
Computation times of the prototype for analysis and reduced model generation	nop		2.96 secs
	read		6.53 secs
	write		3.24 secs
FIFO sizes on RTL	nop	160 cells x 10 bit	160 cells x 2 bit
	read	160 cells x 10 bit	160 cells x 3 bit
	write	160 cells x 10 bit	160 cells x 3 bit
Overall number of bits in all relevant signals (cones of influence of the property)	nop	20925	5034 (24.0 %)
	read	31452	10592 (33.6 %)
	write	14622	5163 (35.3%)
Overall number of gates in synthesized netlist	nop	23801	5661 (27.9 %)
	read	23801	7929 (33.3 %)
	write	23801	7929 (33.3 %)
Number of state bits	nop	1658	362 (21.8 %)
	read	1658	524 (31.6 %)
	write	1658	524 (31.6 %)
Property checker runtimes	nop	23:33 min	37.96 secs (2.7 %)
	read	42:23 min	3:27 min (8.1 %)
	write_fail	2:08 min	25.66 secs (19 .5 %)
	write_hold	27:08 min	1:08 min (4.2 %)

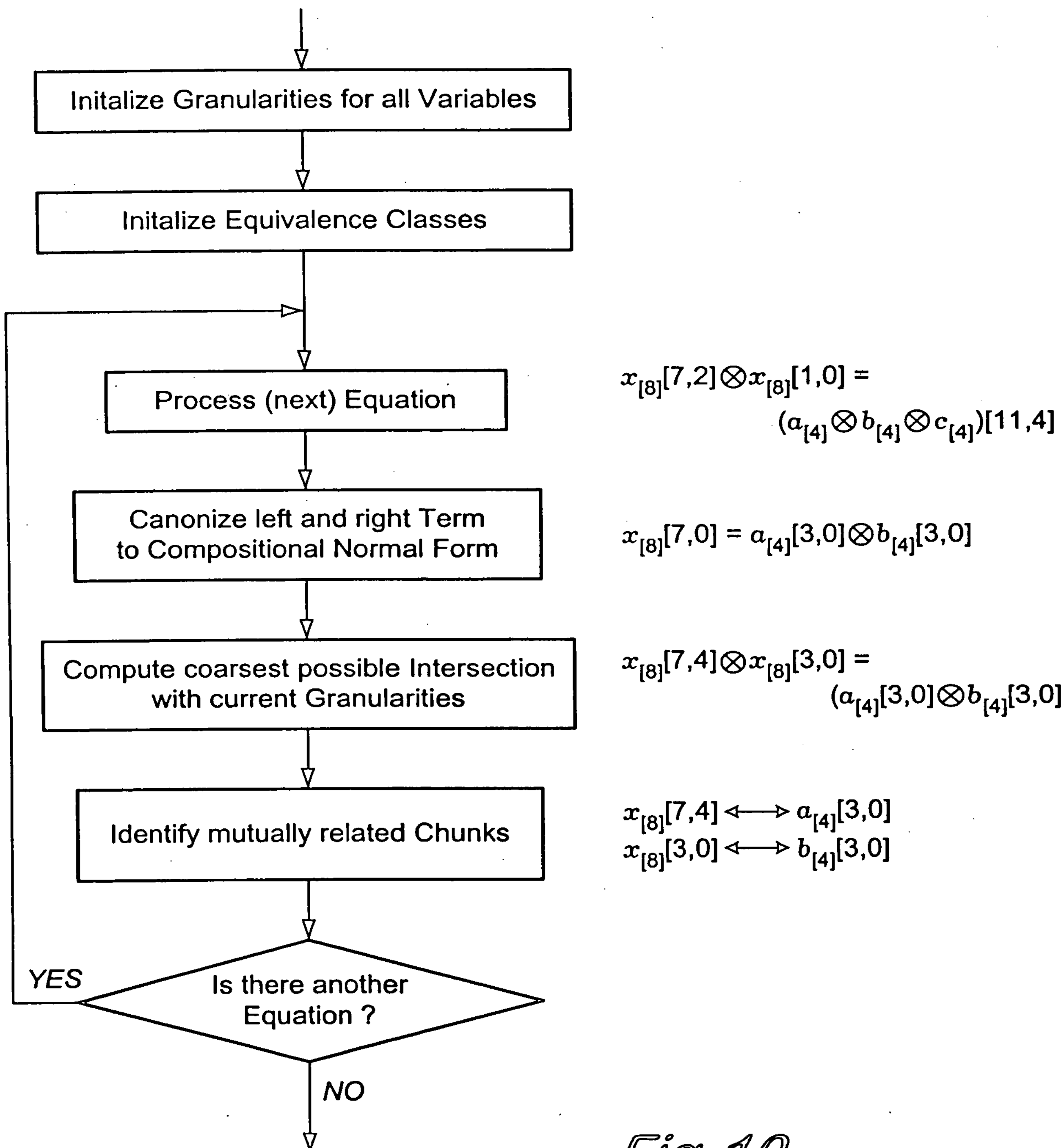
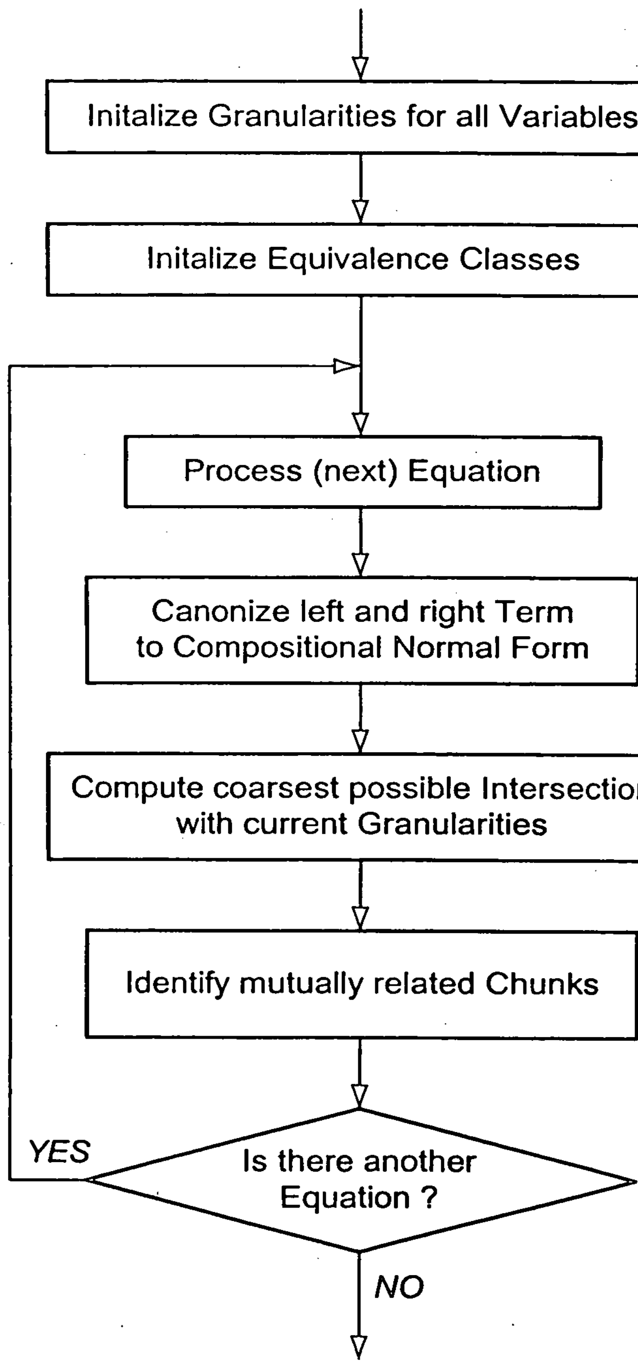

Initalize Granularities for all Variables
Initalize Equivalence Classes
Process (next) Equation
$x_{[8]}[7,2]\otimes x_{[8]}[1,0] = (a_{[4]}\otimes b_{[4]}\otimes c_{[4]})[11,4]$
Canonize left and right Term to Compositional Normal Form
$x_{[8]}[7,0] = a_{[4]}[3,0]\otimes b_{[4]}[3,0]$
Compute coarsest possible Intersection with current Granularities
$x_{[8]}[7,4]\otimes x_{[8]}[3,0] = (a_{[4]}[3,0]\otimes b_{[4]}[3,0]$
Identify mutually related Chunks
$x_{[8]}[7,4] \longleftrightarrow a_{[4]}[3,0]$
$x_{[8]}[3,0] \longleftrightarrow b_{[4]}[3,0]$
Is there another Equation ?
YES
NO

Fig. 10



$x_{[8]} = \mathrm{ite}(a_{[4]} = b_{[4]}, y_{[8]}, z_{[8]})$

$x_{[8]} = \mathrm{ite}(a_{[4]} = b_{[4]}, y_{[8]}, z_{[8]})$

$x_{[8]}[7,0] = \mathrm{ite}(a_{[4]}[3,0] = b_{[4]}[3,0], y_{[8]}[7,0], z_{[4]}[7,0])$

$a_{[4]}[3,0] \longleftrightarrow b_{[4]}[3,0]$
$x_{[8]}[7,0] \longleftrightarrow y_{[8]}[7,0]$
$x_{[4]}[7,0] \longleftrightarrow z_{[8]}[7,0]$

Fig. 11

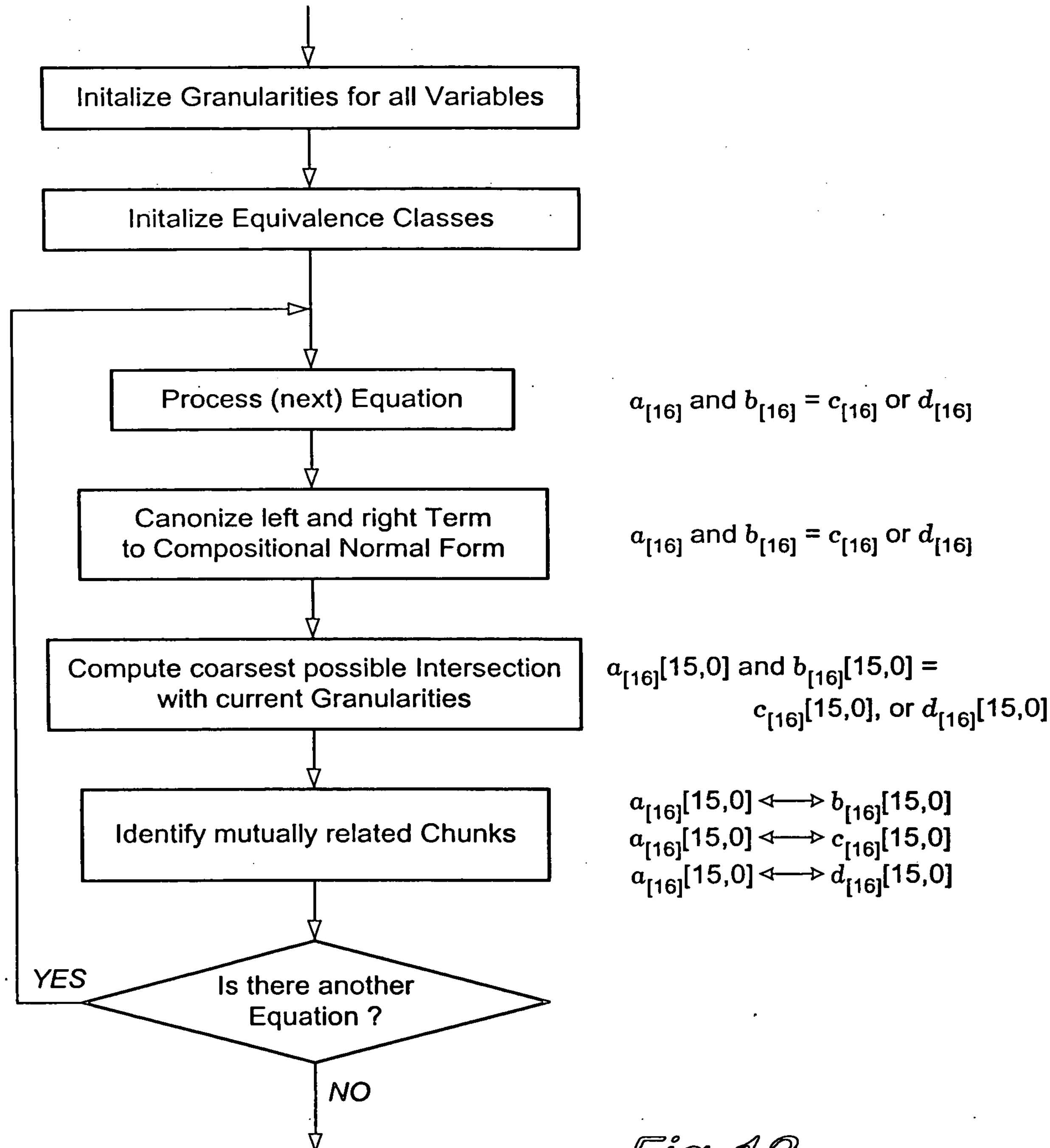

Initalize Granularities for all Variables
Initalize Equivalence Classes
Process (next) Equation
$a_{[16]}$ and $b_{[16]} = c_{[16]}$ or $d_{[16]}$
Canonize left and right Term to Compositional Normal Form
$a_{[16]}$ and $b_{[16]} = c_{[16]}$ or $d_{[16]}$
Compute coarsest possible Intersection with current Granularities
$a_{[16]}[15,0]$ and $b_{[16]}[15,0] = c_{[16]}[15,0]$, or $d_{[16]}[15,0]$)
Identify mutually related Chunks
$a_{[16]}[15,0] \longleftrightarrow b_{[16]}[15,0]$
$a_{[16]}[15,0] \longleftrightarrow c_{[16]}[15,0]$
$a_{[16]}[15,0] \longleftrightarrow d_{[16]}[15,0]$
Is there another Equation ?
YES
NO

Fig. 12

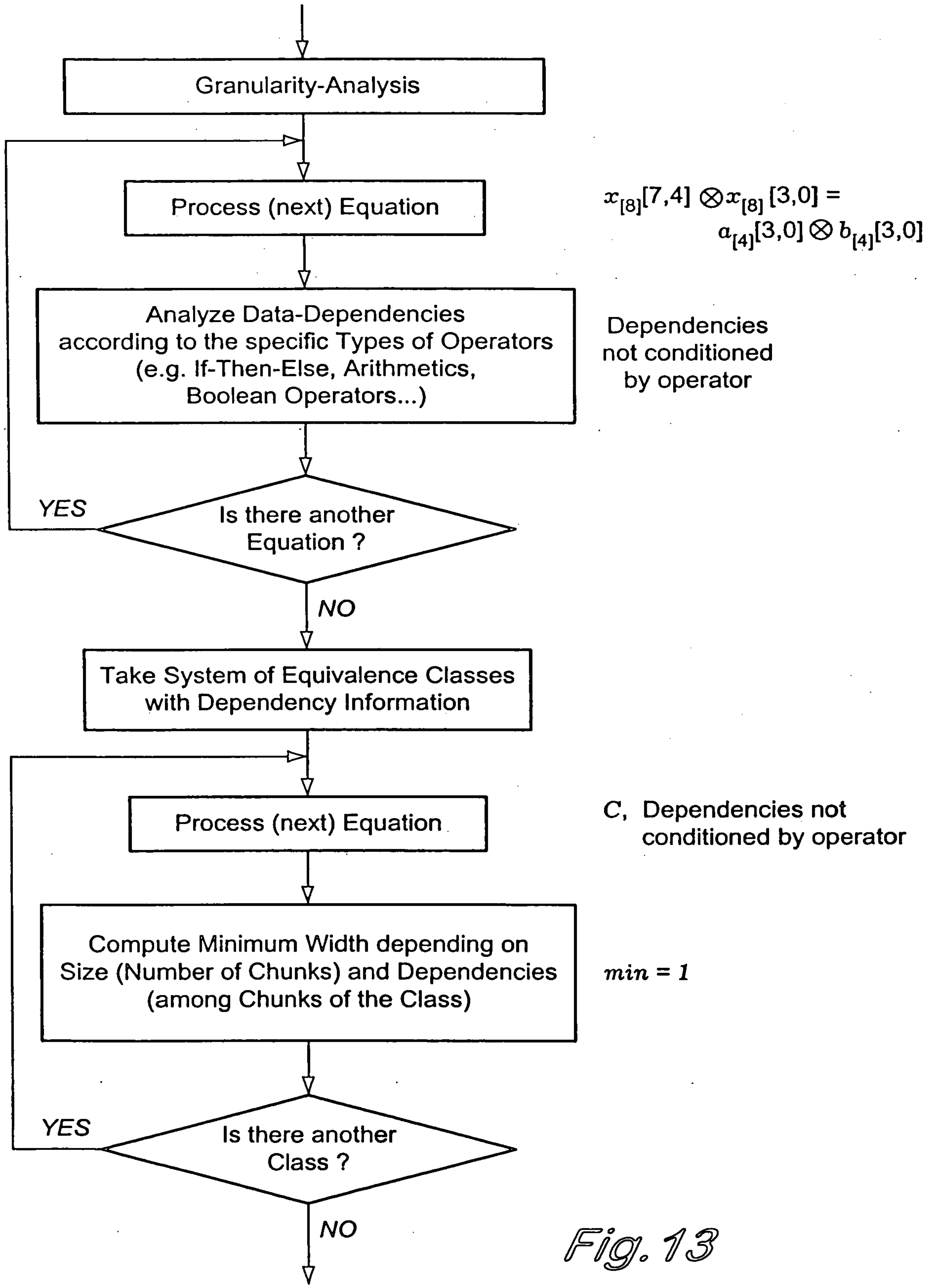

Granularity-Analysis
Process (next) Equation
$x_{[8]}[7,4] \otimes x_{[8]}[3,0] = a_{[4]}[3,0] \otimes b_{[4]}[3,0]$
Analyze Data-Dependencies according to the specific Types of Operators (e.g. If-Then-Else, Arithmetics, Boolean Operators...)
Dependencies not conditioned by operator
YES
Is there another Equation ?
NO
Take System of Equivalence Classes with Dependency Information
Process (next) Equation
C, Dependencies not conditioned by operator
Compute Minimum Width depending on Size (Number of Chunks) and Dependencies (among Chunks of the Class)
$min = 1$
YES
Is there another Class ?
NO

Fig. 13

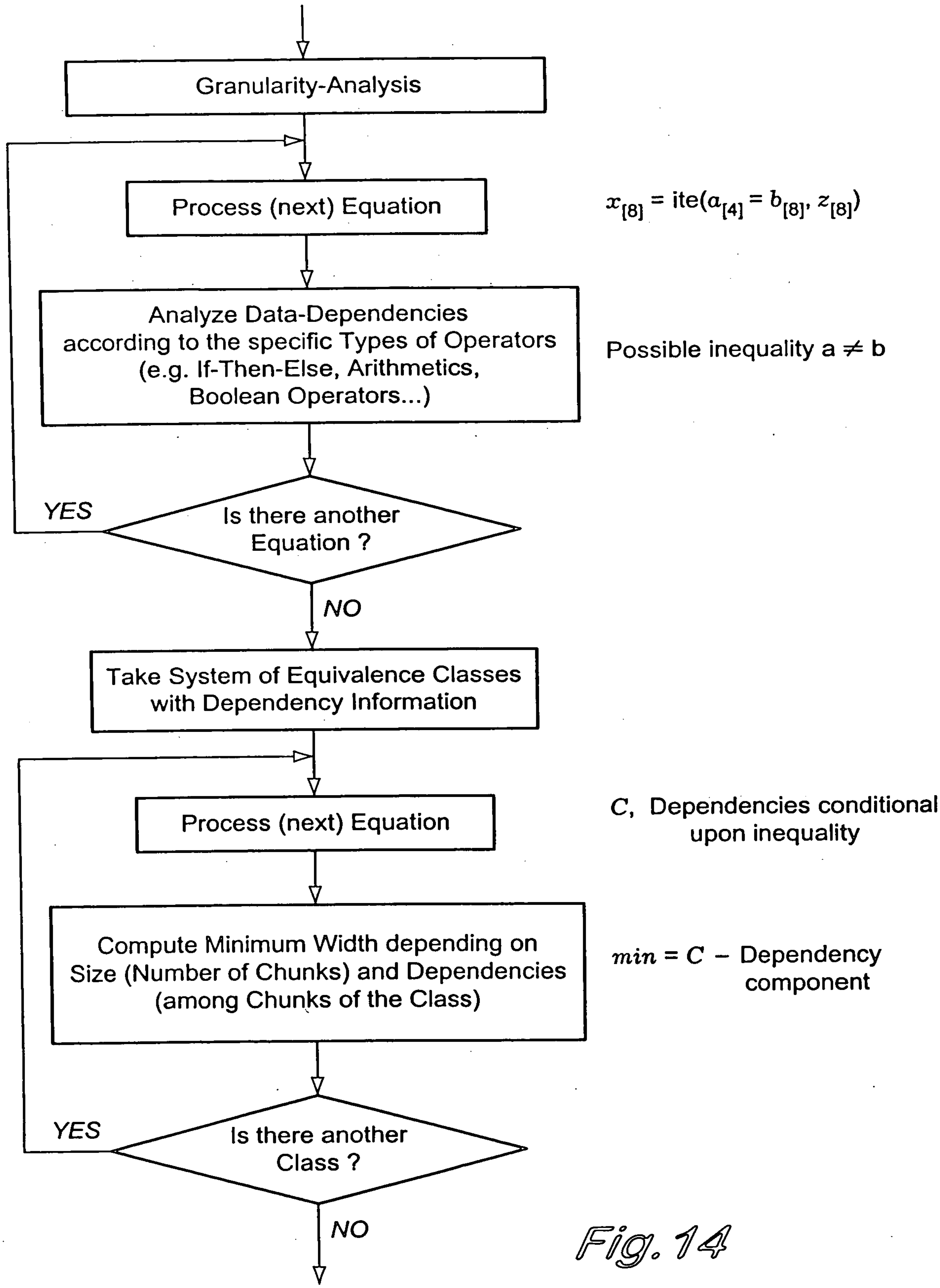

Granularity-Analysis
Process (next) Equation
$x_{[8]} = \text{ite}(a_{[4]} = b_{[8]}, z_{[8]})$
Analyze Data-Dependencies according to the specific Types of Operators (e.g. If-Then-Else, Arithmetics, Boolean Operators...)
Possible inequality a ≠ b
Is there another Equation ?
YES
NO
Take System of Equivalence Classes with Dependency Information
Process (next) Equation
C, Dependencies conditional upon inequality
Compute Minimum Width depending on Size (Number of Chunks) and Dependencies (among Chunks of the Class)
$min = C$ – Dependency component
Is there another Class ?
YES
NO

Fig. 14

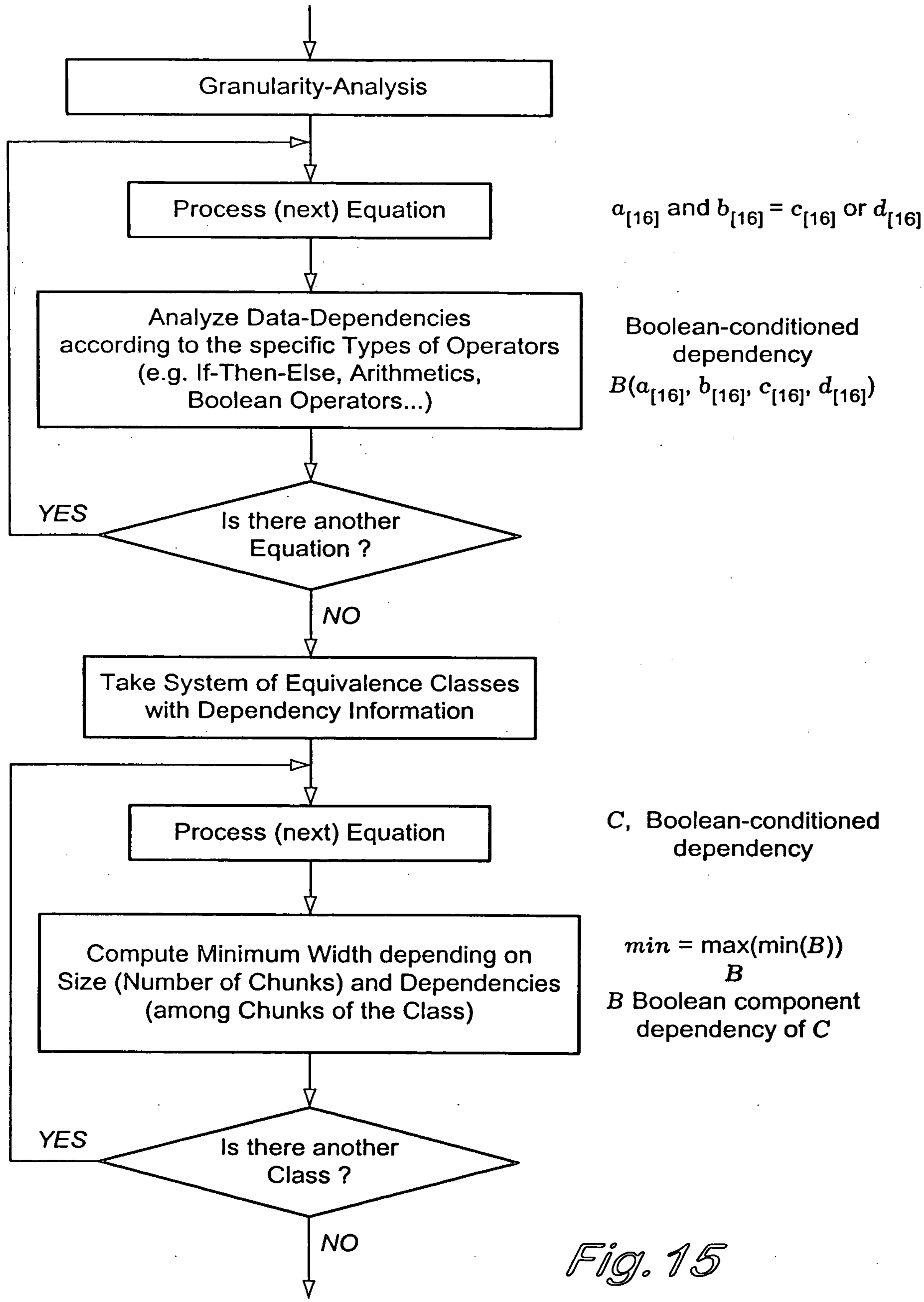

Granularity-Analysis
Process (next) Equation
$a_{[16]}$ and $b_{[16]} = c_{[16]}$ or $d_{[16]}$
Analyze Data-Dependencies according to the specific Types of Operators (e.g. If-Then-Else, Arithmetics, Boolean Operators...)
Boolean-conditioned dependency
$B(a_{[16]}, b_{[16]}, c_{[16]}, d_{[16]})$
YES
Is there another Equation ?
NO
Take System of Equivalence Classes with Dependency Information
Process (next) Equation
C, Boolean-conditioned dependency
Compute Minimum Width depending on Size (Number of Chunks) and Dependencies (among Chunks of the Class)
$min = \max_B(\min(B))$
B Boolean component dependency of C
YES
Is there another Class ?
NO

Fig. 15

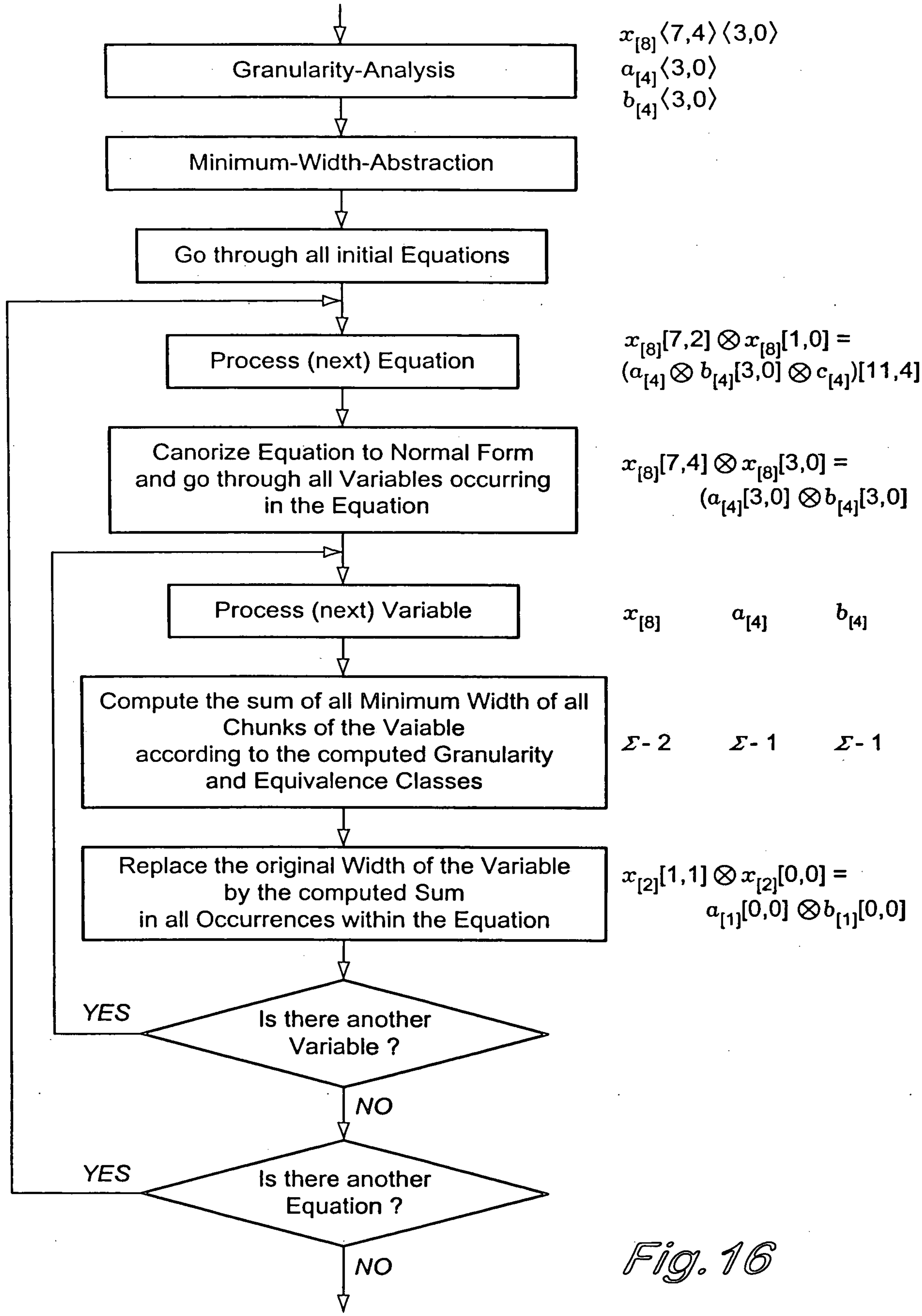

Granularity-Analysis
Minimum-Width-Abstraction
Go through all initial Equations
Process (next) Equation
Canorize Equation to Normal Form and go through all Variables occurring in the Equation
Process (next) Variable
Compute the sum of all Minimum Width of all Chunks of the Vaiable according to the computed Granularity and Equivalence Classes
Replace the original Width of the Variable by the computed Sum in all Occurrences within the Equation
Is there another Variable ?
YES
NO
Is there another Equation ?
YES
NO
$x_{[8]}\langle 7,4\rangle\langle 3,0\rangle$
$a_{[4]}\langle 3,0\rangle$
$b_{[4]}\langle 3,0\rangle$
$x_{[8]}[7,2] \otimes x_{[8]}[1,0] =$
$(a_{[4]} \otimes b_{[4]}[3,0] \otimes c_{[4]})[11,4]$
$x_{[8]}[7,4] \otimes x_{[8]}[3,0] =$
$(a_{[4]}[3,0] \otimes b_{[4]}[3,0]$
$x_{[8]}$ $a_{[4]}$ $b_{[4]}$
Σ- 2 Σ- 1 Σ- 1
$x_{[2]}[1,1] \otimes x_{[2]}[0,0] =$
$a_{[1]}[0,0] \otimes b_{[1]}[0,0]$

Fig. 16

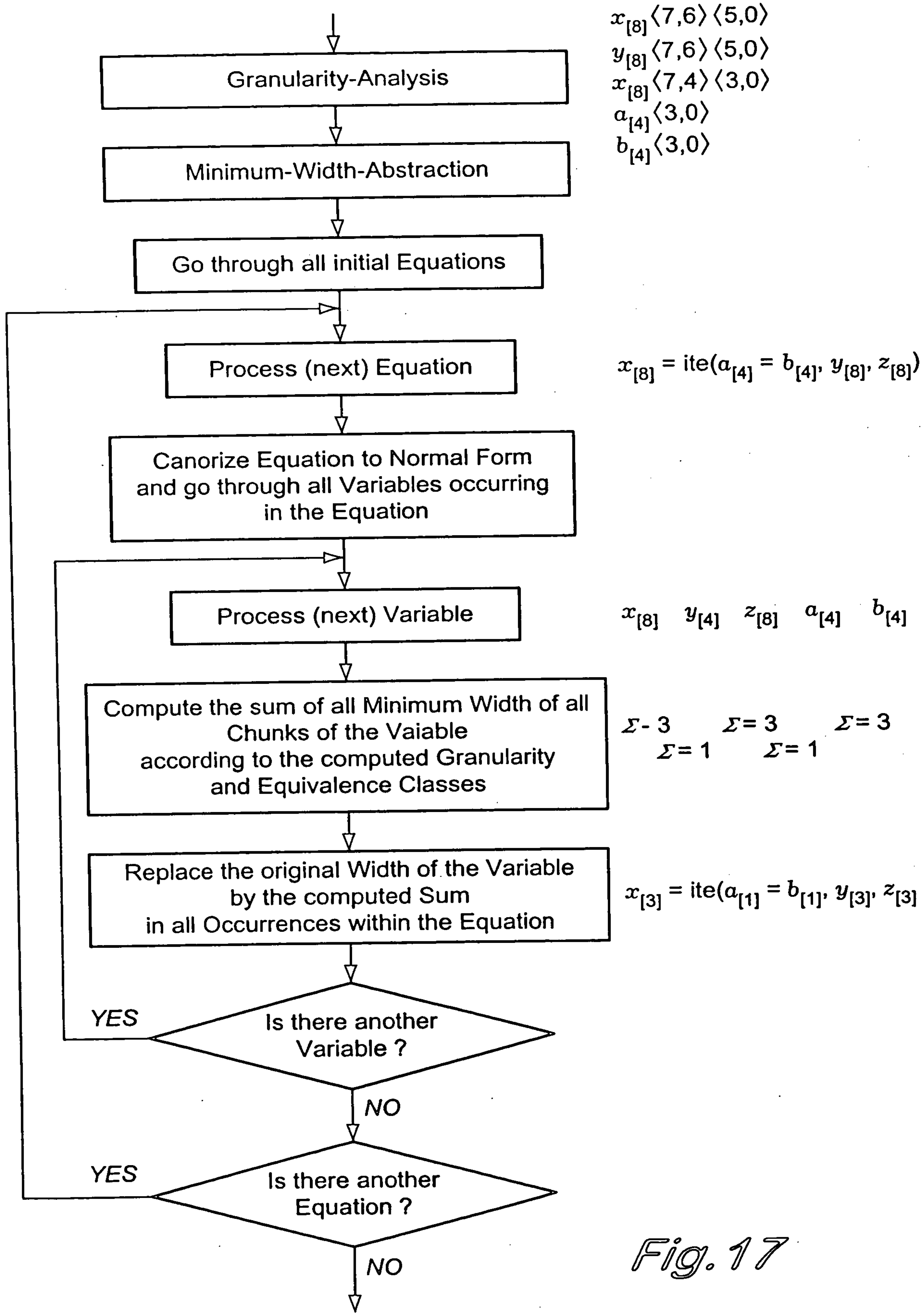

Granularity-Analysis
Minimum-Width-Abstraction
Go through all initial Equations
Process (next) Equation
Canorize Equation to Normal Form and go through all Variables occurring in the Equation
Process (next) Variable
Compute the sum of all Minimum Width of all Chunks of the Vaiable according to the computed Granularity and Equivalence Classes
Replace the original Width of the Variable by the computed Sum in all Occurrences within the Equation
Is there another Variable ?
YES
NO
Is there another Equation ?
YES
NO
x[8]⟨7,6⟩⟨5,0⟩
y[8]⟨7,6⟩⟨5,0⟩
x[8]⟨7,4⟩⟨3,0⟩
a[4]⟨3,0⟩
b[4]⟨3,0⟩
x[8] = ite(a[4] = b[4], y[8], z[8])
x[8] y[4] z[8] a[4] b[4]
Σ- 3 Σ= 3 Σ= 3
Σ= 1 Σ= 1
x[3] = ite(a[1] = b[1], y[3], z[3]

Fig. 17

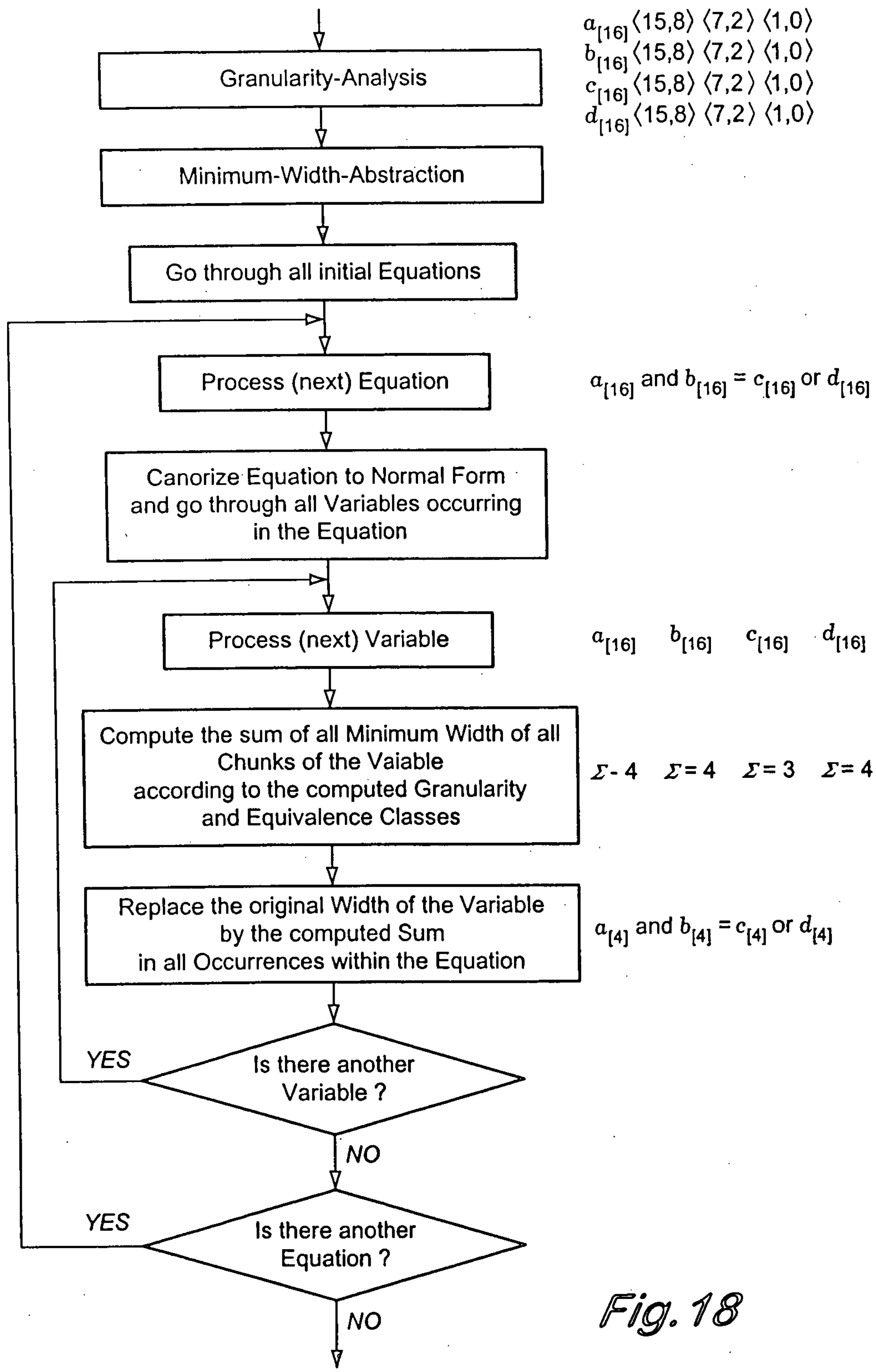

Granularity-Analysis
$a_{[16]} \langle 15,8 \rangle \langle 7,2 \rangle \langle 1,0 \rangle$
$b_{[16]} \langle 15,8 \rangle \langle 7,2 \rangle \langle 1,0 \rangle$
$c_{[16]} \langle 15,8 \rangle \langle 7,2 \rangle \langle 1,0 \rangle$
$d_{[16]} \langle 15,8 \rangle \langle 7,2 \rangle \langle 1,0 \rangle$
Minimum-Width-Abstraction
Go through all initial Equations
Process (next) Equation
$a_{[16]}$ and $b_{[16]} = c_{[16]}$ or $d_{[16]}$
Canorize Equation to Normal Form and go through all Variables occurring in the Equation
Process (next) Variable
$a_{[16]}$ $b_{[16]}$ $c_{[16]}$ $d_{[16]}$
Compute the sum of all Minimum Width of all Chunks of the Vaiable according to the computed Granularity and Equivalence Classes
Σ- 4 Σ= 4 Σ= 3 Σ= 4
Replace the original Width of the Variable by the computed Sum in all Occurrences within the Equation
$a_{[4]}$ and $b_{[4]} = c_{[4]}$ or $d_{[4]}$
Is there another Variable ?
YES
NO
Is there another Equation ?
YES
NO

Fig. 18